RESEARCH ARTICLE

Reverse engineering large-scale genetic networks: synthetic versus real data

LUWEN ZHANG[1,2], MEI XIAO[1,2], YONG WANG[3] and WU ZHANG[1]*

[1] *School of Computer Engineering and Science, Shanghai University, 149 Yanchang Road, Zhabei District, Shanghai 200072, People's Republic of China*
[2] *Institute of Systems Biology, Shanghai University, 99 Shangda Road, Baoshan District, Shanghai 200072, People's Republic of China*
[3] *Academy of Mathematics and Systems Science, Chinese Academy of Sciences (CAS), 52 San Lihe Road, Xicheng District 100864, Beijing, People's Republic of China*

Abstract

Development of microarray technology has resulted in an exponential rise in gene expression data. Linear computational methods are of great assistance in identifying molecular interactions, and elucidating the functional properties of gene networks. It overcomes the weaknesses of *in vivo* experiments including high cost, large noise, and unrepeatable process. In this paper, we propose an easily applied system, Stepwise Network Inference (SWNI), which integrates deterministic linear model with statistical analysis, and has been tested effectively on both simulated experiments and real gene expression data sets. The study illustrates that connections of gene networks can be significantly detected via SWNI with high confidence, when single gene perturbation experiments are performed complying with the algorithm requirements. In particular, our algorithm shows efficiency and outperforms the existing ones presented in this paper when dealing with large-scale sparse networks without any prior knowledge.

[Zhang L., Xiao M., Wang Y. and Zhang W. 2010 Reverse engineering large-scale genetic networks: synthetic versus real data. *J. Genet.* **89**, 73–80]

Introduction

Systems biology studies biological systems by systematically perturbing them genetically. The scope of systems biology is monitoring the elements of biology, integrating various levels of data, and, ultimately, modelling the biological process computationally to describe the structure of the system and reconstruct the molecular networks. An enormous challenge for systems biology is how to simulate the complex biological system by efficient computer modelling tools. Such tools are gaining importance as a paradigm shift is occurring in biology away from a descriptive science toward a predictive one, along with large-scale technologies beginning to generate vast amounts of quantitative data (Faith *et al.* 2007).

Computational models and computer-aided tools have already achieved widespread acceptance within engineering science and bioinformatics fields. In practical research, it is difficult, both for experiments and computation, to reveal all the elements in a gene network because of the complicated experimental design, high noise, and unrepeatable process (Styczynski and Stephanopoulos 2005; Margolin and Califano 2007; Schumacher *et al.* 2007). Other considerable difficulties are addressed as: firstly, the number of gene expression profiles available is much less than the number of genes which can be abstracted to ill matrix problem; secondly, there is lack of sufficient valuable previous knowledge to help choosing motifs and central regulators; thirdly, the linkages between genes have not been well defined so generally the gene interactions detected by inferring methods are not physical interactions but the influent interactions. Due to the different mathematical formalisms used to model net-

*For correspondence. E-mail: wzhang@shu.edu.cn.

Keywords. gene regulatory network; single gene perturbation; linear model; stepwise; simulated network.

works, there is finite overlap with each inferred result. Till now, there is no standard model describing the regulatory mechanism for the genes and even the post-transcriptional modification. Many mature theories and models from systems area cannot be simply applied to gene network reconstruction because of poor data, thus also making the problem indeterminate.

Recently many methods integrating computer science and statistics have been applied to infer the underlying gene network via microarray expression profiles, including Boolean networks (Liang *et al.* 1998), Bayesian networks (Friedman *et al.* 2000; Beal *et al.* 2005), relevance networks (Butte and Kohane 1999), graphical models (De la Fuente *et al.* 2004), genetic algorithms (Iba and Mimura 2002), neural networks (van Someren *et al.* 2001), controlled language -generating automata (Chen 2004), linear ordinary differential equations (Yeung *et al.* 2002), and non-linear differential equations (Mendes *et al.* 2003). All these models offer unique advantages in many aspects in gene regulatory network (GRN) construction, although they do also have their shortcomings. For example, Boolean networks use the discrete variable model to approximate gene states as either ON or OFF, which is unable to capture some behaviours of gene circuits. Bayesian models, however allow a natural way to select one model from a set of competing ones to best describe the experimental data, easily leading to an NP-hard problem. Hence, these cause significant increase in computing complexity in learning the network architecture. Based on language-generating automata, Chen (2004) designed an alternative model which regards each gene as an automaton and the pathway can be computed by the intersection of languages; however this approach has not yet been tested on real experimental data.

Linear models use a set of ordinary differential equations to describe a gene regulatory system, in which each gene is influenced by all the others. Such models can conveniently represent continuously changing gene expression values. The mRNA concentrations measured from microarray experiments and global or local perturbations to the gene pool are put into the linear model to generate an $N \times N$ matrix, where N is the number of genes. Regulations can be weighted with sign that distinguish active and repressed regulatory states of genes. Because there are far more genes than experiment samples, the gene expression data sets become ill-posed. As a result, there is no exclusive solution to the equations and the task to confirm $N \times N$ parameters seems daunting. In other words, the scarcity of time-course or steady-state data is a major difficulty of GRN inference for all methods (Zak *et al.* 2003). Theoretically, this problem can be overcome by increasing time points and integrating multiple microarray data sets from many public databases (Wang *et al.* 2006). On the other hand, if the number of genes comprising the network can be reduced to be generally equal to the experimental samples, the ill-posed problem may be corrected and a more stable GRN can be reconstructed. It is worth mentioning that complicated experimental design with precise computational modelling to pick valuable genes or regulatory modules typically lose scalability and systematic view on the network. Hence, current GRN inference methods do not perform well with regard to either simplicity or accuracy.

In this paper, we develop a rapid and scalable method for reconstructing GRN using steady-state gene expression measurements without any prior information about gene functions or network structure. We determine the first-order model from expression changes resulting from a set of different transcriptional perturbations. Based on the multiple stepwise linear regression model, we generate a gene network that is sparse and consistent with biologically plausibility, while conventional approaches often derive densely connected regulatory relationships among nodes. Our method, called stepwise network inference (SWNI), has distinct advantages specifically in detecting large-scale gene networks. Further, inferring GRN is modelled as matching and optimizing the possible regulated subsets, thereby a reliable and consistent network structure can be expected. Both simulated and experimental data sets are used to testify the biological effectiveness and computational efficiency of SWNI method.

Methods

General model for gene networks

The dynamics of a genetic network in perturbation can be expressed by a set of ordinary differential equations (De Jong 2002):

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x, p) - s(\ddot{e}). \tag{1}$$

This expression describes the time evolution of the mRNA concentration of the genes in the network. Vector x represents the expression level of the genes, vector p is a set of transcriptional perturbations exerted on genes and $\ddot{e}$ describes the self-degradation rate; these are all row vectors. The perturbation should be small enough (typically 10% of the original mRNA concentration) to ensure that the system can return to the original steady state point. In other words, the network should not be driven out of the basin of attraction of the stable steady state point. With these assumptions, the gene regulatory process can be approximated by a linear system near the steady state point. Thus, for a genetic network consisting of N genes we have:

$$\frac{\mathrm{d}x}{\mathrm{d}t} \sum_{j=1}^{N} w_{ij} x_j - \lambda_i x_i + p_i i = 1, \ldots, N, \tag{2}$$

Where x_is are the mRNA concentrations of gene i; w_{ij}s are the weighted strength of the influence of gene j on gene i, λ_is describe the self-degradation rate of gene i, and p_is are the external perturbation to the expression of gene i. If repeating perturbation M times, for each gene in each experiment l we can rewrite the above equation as:

$$\frac{\mathrm{d}x_{il}}{\mathrm{d}x} = \sum_{j=1}^{N} w_{ij}x_{jl} - \lambda_{il}x_{il} + p_{il}, l = 1 \ldots M, \tag{3}$$

For simplicity, we absorb the self-degradation rate $\lambda_{il}s$ into coupling the weighted strength w_{ij}s and get a_{ij}s, yielding

$$\frac{\mathrm{d}x_{il}}{\mathrm{d}x} = \sum_{j=1}^{N} a_{ij}x_{jl} + p_{il}, i = 1 \ldots N, l = 1 \ldots M. \tag{4}$$

Then we use matrix notation for compaction, yielding:

$$\frac{\mathrm{d}x_l}{\mathrm{d}t} = Ax_l^T + P_l^T, l = 1 \ldots M, \tag{5}$$

where x_l is an $1 \times M$ vector of measured mRNAs in M different experiments, A is an $N \times N$ weighted connectivity matrix composed of $a_{ij}s$ which is unknown. Therefore, inferring GRN is transformed to retrieving matrix A.

Linear regression solution

As analysed, the gene expression level is measured at steady state, so,

$$\frac{\mathrm{d}X}{\mathrm{d}t} = 0. \tag{6}$$

It seems that for the system of equation $AX + P = 0$ where X and P can be measured from experiments conveniently, the solution is obtained simply by inverting X if there exist N experiments for N genes:

$$A = -PX^{-1}. \tag{7}$$

However, there exist two main problems on retrieving a reliable solution (Zak *et al.* 2003). The first one is, typically, the number of experiments is fewer than gene numbers ($M \leq N$) because of costly experiments. The other problem is that A is extremely sensitive to noise in both of the measured data X and P, so we can confirm that even if $M \geq N$, the solution by (7) is unstable and unreliable.

To circumvent those problems, we can assume that the maximum number k of regulators acting on each gene is less than M, thus the number of weights a_{ij} will be reduced. A multiple linear regression method (Gardner *et al.* 2003) tries to identify which combination of k out of N genes is selected as regulatory inputs for gene i by computing the sum of deviations squared for all possible combinations. For each gene, the solution minimizing the least square error is selected.

$$\mathrm{MinSSE}_i^l = \mathrm{Min} \sum_{l=1}^{M} (-p_{il} - \widetilde{a}_i x_l^T)^2, \tag{8}$$

and

$$\widetilde{a}_i^T = (Xx^T)^{-1}X(-p_i^T). \tag{9}$$

The above process contsists of two steps: the first one is identifying the variable k, and the second one is identifying k, out of N regulatory inputs. Unfortunately, the strategy of regulatory inputs has limitations in both of quantity and quality. Simultaneously, the test of statistical significance lacks reliability.

SWNI solution

The major inadequacy of current regression method is how to identify the value of k, which is the maximum regulatory relations for each gene. In order to fix on the maximum subset size, for example, for all the $k = (1, \ldots M)$ (M is the number of experimental samples), the NIR algorithm computes all the inferred genetic networks (regression models) in which each row contains k nonzero genes (regressors) respectively. In fact, it makes a traversal of linear regression $2^M - 1$ ($C_M^k = C_M^1 + C_M^2 + \cdots + C_M^M$) times for each gene and is computationally impossible for large networks. It is computationally costly even to make a simple traversal, which should be avoided. Then for all regression models, the NIR method intends to choose the one with smallest SSE and best significant fit. However, with the increase in k (the number of regressors), the SSE becomes smaller consecutively, which is not expected to be. From this phenomena, when $k = M$, the network model fits best, clearly an undesirable property. Therefore, the identification of k is still a problem that seems difficult to figure out based on quantificational analysis. With qualitative analysis, such as the dynamic stability of networks, balance between coverage and false positives produces a relatively suitable k. One unreasonable result generated from this strategy is that every gene set has to be acted upon by the same number of regulators, k. Thus, if in the real genetic network there are more than k regulatory inputs to certain genes, some of them will be chosen before regression. Moreover, the common regression method ignores the significance test on individual regressor variables that is crucial for eliminating those dispensable regressors from a regression model and keeping the important ones, although it applies significance test on the holistic model to determine if there really exists a linear relationship between the dependent variable and the regressor variables. The typical method applies significant test as a kind of afterwards validation which can not work on the existing fact. In contrast, our method SWNI uses F test and partial F test to guide the selection of predictor variables in the process of regression which makes the results more reliable.

The underlying idea of SWNI is: for each gene, we select the connectivity with the highest probability, one by one. Besides overall F test, the partial F test is utilized to evaluate whether the selected individual regulatory input provides the best fit to the regression model. Every time we add a new relationship, it is necessary to repeat the partial F test upon the old ones to ensure they also have statistical significance with the new model. The process continues until there are no more new regulators that can be added, and no selected regulatory relationships that can be deleted. Finally, we establish the stepwise linear regression model within all selected relationships to calculate the correlation coefficients between each gene and its regulators. We summarize our iterative procedure as follows:

Step 1, choose the first regulatory inputs for gene i.

First, for $-P = AX$, let $-P = Y$ for convenience. Assume that gene i is regulated by only one gene $j\,(j = 1, \ldots N)$. We need to choose one from N regression equations including one predictor variable only.

$$\hat{y}_i = a_{ij}^{(1)} x_j^{(1)}, \tag{10}$$

where $\hat{y}_i$ is an estimate of y_i. The coefficient a_{ij} is the weighted influence of regulator j to gene i. Next for $j = 1 \ldots N$, the predictor is selected by F test (P value < 0.03) to get the regression coefficients. Then compute,

$$F_j^{(1)} = \frac{\Delta\,\mathrm{SSR}_j}{\mathrm{SSE}\,(N-k)}, \tag{11}$$

where ΔSSR is the partial sum of square due to the regression of x_j, SSE is the sum of square error due to the linear regression of y to x, k is the number of connectivity (in step one $k = 1$), and N-k is the degree of freedom. For the significant level $\alpha = 0.05$, if $F_j^{(1)} > F_{1-\alpha}\,(1, n-1)$, the first connection can be selected into the model. Otherwise, no connection is selected in step one.

Step 2, choose the second regulatory inputs for gene i.

Based on step 1, we make another choice from N-1 regulators for gene i. The following equation includes the first selected gene and the second possible connection.

$$\hat{y}_i = a_{ij}^{(1)} x_j^{(1)} + a_{ij}^{(2)} x_j^{(2)}. \tag{12}$$

Next, the partial F test is applied to the second possible predictor. For the significant level P value, if $\alpha_{in} < \alpha_{out}$, the second gene can be selected. It is important that if the second gene is chosen, the first one must be evaluated by partial F test. But, if for the significant level $\alpha_{out} = 0.1$, $F_j^{(1)} < F_{1-\alpha}(1, n-2)$, it will be eliminated. Here we should consider carefully that α_{in} must be different from α_{out}, generally set $\alpha_{in} < \alpha_{out}$ or the closed cycle loop will be easy to generate, then the predictor will be endlessly selected and deleted, which is not desirable.

Step 3, the condition for stopping selection.

We continuously select regulators following the rules in step 2. If the selected variables are always significant fit for the model, a new selection will be considered. At the end, the model may include k variables which can not be eliminated and there is no regulator that can be selected into the model. Thus, the final regression model for gene i is:

$$-P_i = \widetilde{a}_i X. \tag{13}$$

Where p_i is a $1 \times M$ vector for the expression level of gene i in total M experiments $\widetilde{a}_i$, is a $1 \times k$ vector representing the influence coefficient of k selected confident regulatory inputs j to gene i, and X becomes a $k \times M$ matrix.

Experiments and results

In this section, the performance of our method is first evaluated using gold standard networks with random scale-free structure by varying the network size, and average degrees. Because the mechanisms of simulated gene networks are completely known, we are able to faithfully evaluate the prediction result of our algorithm. However, the model used for generating data is actually a simplification of real molecular networks, and this might lead to systematic deviations. Meanwhile, the shortcoming of biased evaluation can also be addressed using only real steady-state expression data. By combining both assessment approaches, we are likely to obtain a more reliable picture of the performance of the algorithm. Next, we will describe in detail how to use synthetic data and real experimental data to evaluate our method.

Application to synthetic data

Apart from manual design of some small benchmark networks, three classes of directed networks are currently used as models for generating *in silico* gene regulatory network structures: random (Kauffman 1974), 'small-world' (Watts and Strogatz 1998) and scale-free (Albert and Barabasi 1999, 2000). Networks with scale-free topology are perhaps the best suited for simulating GRNs, though it may still be controversial, we opt to construct the model in scale-free class. There are some evidences that metabolic networks display properties similar to scale-free networks. Even at the level of gene networks, this similarity also exists, for instance, based on expression profiles of yeast mutants studied by (Featherstone and Broadie 2002). According to their connectivity in networks, the distribution of vertices follows a power law that the minority of vertices have a very large number of connections, while a large number of vertices have only a few connections (Mendes *et al.* 2003).

Instead of constructing more complicated graph models, which would be unfair in assessment, we believe that the fairest way to compare reverse engineering methods is based on real biological network structures. Following the scale-free topology, 10 'gold standard' networks of size 50 were generated using an *Arabidopsis thaliana* transcriptional regulatory network (Mendoza *et al.* 1999) as source. Meanwhile, the generated networks must be stable which can be testified by the eigenvalues of the corresponding matrixes. The dynamics are asymptotically stable if real parts of all eigenvalues are negative or the largest eigenvalue has negative real part. The more negative the leading eigenvalue is, the faster the system returns to equilibrium following small perturbations (Chen and Aihara 2002).

For each stable system (simulated network), steady-state data were generated resulting from M local perturbations by solving (7) for X, and adding an error term $E : X = -A^1 P + E$. To be more precise, in each numerical experiment the expression level of a different single gene was increased at the same rate, because for the linear model, the size of the perturbation is not important and it was set to 10% in this case. Normally distributed noise, with zero mean and standard deviation which was multiplied by 1/10th the absolute value of the simulated gene expression level was also added. Where A is the simulated network, P is a diagonal matrix that is set to identity and E can be seen as experimental error, and the effects of nonlinearity of gene regulations. In addition, the noise E is added proportional to the size of each element, since large values will receive more absolute noise than low ones.

In order to evaluate the accuracy of our predictive model, we also justified the performance of various alternative algorithms listed in table 1 and facilitated comparison among them. All the algorithms were implemented in MATLAB 7.0 and run on all the 'gold standard' data sets using default parameters: fifty-node scale-free networks with average degree 2.9. We averaged the results over 10 data sets that were generated from the 10 gold standard networks.

We compared simulated test networks with the predicted networks via the algorithms and present in a structure known as a confusion matrix. The standards presented in table 2 (b) can be used to construct a point in either receive operating characteristic (ROC) space or precision-recall (PR) space. Both of the two curves are useful for presenting results for binary decision problems and measuring the quality of the network reconstruction in this study. ROC curve can illustrate an overly optimistic view of the performance of the algorithm by describing the trade-off between sensitivity and the false positive rate. However, for simulated networks here, which have similar topology to real GRNs, which are generally sparse, the ROC curve often suffers from a high false positive rate. The PR curve instead is based on computing precision and recall (true positive rate), and therefore gives a more accurate picture when dealing with a highly skewed data set (Davis and Goadrich 2006). Moreover, looking at the PR curve can highlight differences between algorithms that are not apparent in ROC space. The goal in ROC space is to be in the upper-left-hand corner, the more, the better. However, PR curve in the upper-right-hand corner of the space is considered good.

Results of the application of six network inference algorithms on the same generated data set are described in figure 1. PR curve and ROC are displayed in two columns. It seems that the two approaches, SWNI and NIR, which are built on the basis of linear ordinary differential equations, significantly outperform the others in this case for larger area under the ROC curve they occupy. SWNI recovers almost all the real directed interactions (high sensitivity and high precision) and has visible improvement compared to NIR which covers most of the true connections in the network with high sensitivity and little decrease in accuracy. At the same time, because of only few detected true edges, BANJO and clustering failed, as their performance is comparable with the random model (experiential $P \leq 0.1$). ARACNE have large space in improvement of TPR and Precision. When observing the ROC and PR curve, the algorithms are similar ranged by their performance, in agreement with the results of Davis *et al.* (2006) that a curve dominates in ROC space if and only if it dominates in PR space. Moreover, it is important to remark that the prediction coverage listed in table 1 shows small gap between the algorithms compared to the ROC curves. As a subclass of two kinds of evaluation standards: accuracy (including root mean squared error, ROC curves, precision-recall, etc.) and usefulness (including coverage, confidence, etc.), coverage been used only can not estimate the reconstruction method roundly.

Further, in order to test the performance of the SWNI method on networks with different size and sparseness, 100 random scale-free networks with an average degree 1 was generated of sizes 10, 20, 50, 100, 200, 500, respectively. We also generated another 100 test networks with an average degree 2.4, 2.7, 4.4, 4.7, 6.5, 7 corresponding to network sizes 10, 20, 50, 100, 200 and 500.

In figure 2, the reconstruction results of all the 200 artificial networks with SWNI is shown for different network sizes in different average degrees, which reveal the statistical property of the system. In this study, the area under the curve (AUC) of both quantities in ROC and PR will be used to give a compact description for varying number of genes. By comparing the two lines in AUC (ROC) space in figure 2, it is possible to examine the influence of average degree on the performance of our algorithm. An AUC (ROC) close to 0.5 corresponds to random, and higher than 0.8 is good. Under equal conditions (network topology and gene numbers), SWNI performed better for sparser networks, and the AUC (PR) also confirm that. If we focus on another network parameter, we can find that the method performance is also influenced by the network size. SWNI performs well in all conditions, and significantly improves with increase of network size from 10 to 200, then tends to stabilize. However, it is interesting that while AUC (ROC) keeps growing with the

Table 1. Features of the gene network inference algorithms and prediction coverage on our data set.

Algorithm	Class	Type of data	Prediction coverage
SWNI	Ordinary differential equation	Steady-state	87%
ARACNE (Basso *et al.* 2005)	Information-theoretic approach	Steady-state/time-series	84%
Clustering (Amato *et al.* 2006)	Hierarchical clustering	Steady-state/time-series	63%
NIR	Ordinary differential equation	Steady-state	85%

Table 2. Common algorithm evaluation metrics.

(a) Confusion matrix

	True edges	Zero edges
Predicted true	TP	FP
Predicted Zero	FN	TN

(b) Definition of metrics

True positive rate (TPR/sensitivity/recall)=TP/(TP+FN)
True negative rate (TNR/specificity) = TN/(TN+FP)
False positive rate (TPR) = FP/(FP+TN)
Precision = TP/(TP+FP)
Coverage = TP+TN/(TP+TN+FP+FN)

increase of gene number until 200 genes, the contrary trend can be observed for AUC (PR). The reason is that along with the rise of network size and sparsely, our algorithm can reach higher sensitivity (large amount of correctly predicted true edges, small false predictions on true edges) and specificity (many correctly inferred zero edges, and few spurious edges) at the cost of reducing precision (too high FP relative to TP).

Application to real steady-state gene perturbation data

We also collected steady-state with single gene perturbation microarray data sets in table 3 for evaluation purpose in two organisms: *Arabidopsis thaliana* and *Escherichia coli*. The 'gold standard' of known regulatory interactions in the two regulatory networks was used to assess the performance of the algorithms. However, the real data sets are not suited for individually comparing between our algorithms and the others, owing to the limited number of data and the imperfect mechanical knowledge of the real network. Analysis performed *in silico* in the previous section is better suited for this task. Here, we still chose a ten-gene network controlling flower morphogenesis in *A. thaliana* proposed by Mendoza *et al.* (1999), and a nine-gene transcript sub-network of the SOS pathway in *E. coli* controlling cell survival and repair after DNA damage as the test networks (Kauffman 1974).

Figure 3 presents the performance of the three gene network reconstruction models, SWNI, NIR and regulatory strengths (RS) (De la Fuente *et al.* 2002), using the available experimental data. When we look at the specificity (figure 1), the SWNI approach outperforms the other two algorithms for reaching nearly 97%. Meanwhile, the RS approach get zero specificity because it does not 'select' regulators but set all genes connected (no true negatives). As for the performance precision of the algorithms, SWNI are significantly better than both NIR and regulatory strength since it tends to choose the most confident connections and has high positive predictions with low-negative predictions as well.

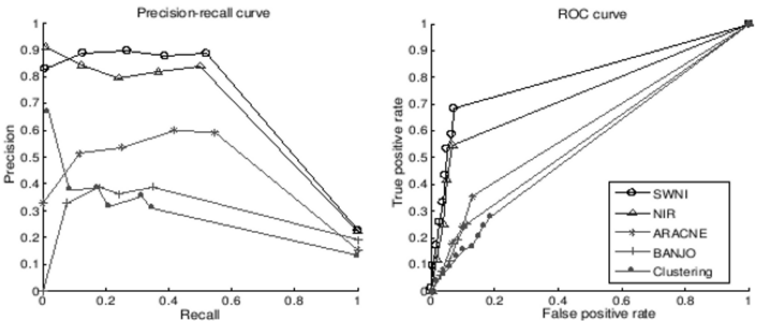

Figure 1. Evaluating the reconstructions results on the gold standard networks of 50 genes with average degree equals to 2.9.

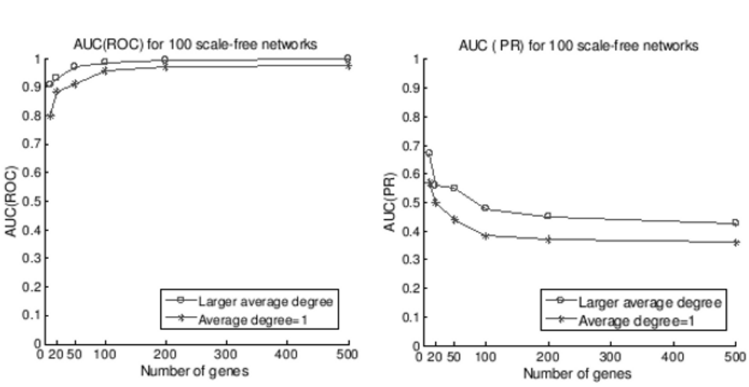

Figure 2. Performance of SWNI applied to 100 scale-free networks with different network sizes and average degree.

Table 3. Real GRNs used as test networks and the experimental data sets.

Organism	Type of data	Number of genes	Average-degree	True edges	Zero edges
Arabidopsis thaliana	Local perturbation, Steady-state	10	2.9	29	71
Escherichia coli	local perturbation, Steady-state	9	5.4	49	32

For instance, SWNI predicts 11 connections in the *E. coli* sub-network and 10 of them are true, while NIR predicts 45 edges, of which only 28 really exist. On the other hand, by setting too high a threshold of selecting significant gene interactions, SWNI unavoidably ignores many secondary important connections and causes low sensitivity on both data sets (many false negatives). Furthermore, higher noise levels in the real data sets than the simulated ones, or imperfect knowledge of the real gene network, may also affect the sensitivity of SWNI. When focusing on the overall performance (specificity, precision and coverage) of SWNI, we found that the results relative to the first network with smaller average degree is considerably better than the second, in line with the *in silico* experimental results. Although the two test data sets we used in this case are not very informative, since the corresponding test networks are small and densely connected, SWNI still performs satisfactorily.

Discussion

We have proposed a rapidly inferring method SWNI based on linear regression and tested our approach on both simulated data sets and experimental gene expression data, which verified the efficiency and effectiveness of the algorithm. SWNI is a deterministic algorithm based on linear regression, and if the noise on the data is not more than 20%, it does not require large data sets for reaching high sensitivity and accuracy. It applies significance test not only on the holistic model to determine if there really exists a linear relationship between the dependent variable and the regressor variables, but also on individual regressor variables that is crucial for eliminating those dispensable regressors from a regression model and keeping the important ones (Shieh *et al.* 2008). The performance of SWNI can be further considerably improved, both algorithmically and in terms of modelling. The novelties and merits of SWNI are summarized as follows:

First, the predictive power is higher for sparse networks than dense ones (Soranzo *et al.* 2007), in particular via SWNI algorithm. When the network becomes larger and sparser, the precision will drop with increase in true positives and coverage. As expected, there are many spurious edges relative to correctly prediction positives, though they are not meaningful to correctly predicted negatives, because that most of the elements in the sparse weighted matrix are zero.

Second, a linear computational model in this paper is used to reconstruct GRN. Although reverse engineering nonlinear dynamical systems with large noise using linear model is a notoriously difficult problem, it has been demonstrated that the use of a linear mapping leads to the efficiently correct deduction of the connectivity of an underlying nonlinear behaviour (Gardner and Faith 2005). As for the network generated from real data sets with SWNI, we can find that our novel approach is robust to nonlinear behaviour. Of course, it may be because structural perturbations we used in this study are more efficient than dynamic perturbations for the purpose of nonlinear system prediction. Nevertheless, SWNI infers a remarkably high number of true edges comparing with the NIR and RS algorithms. Most positive predictions are true, though not every true edges of the network are recovered (many more true positives than false positives). Predictions with high specificity and precision about gene interactions may further give insights about gene pathway. Another advantage of such a linear strategy is that the model can capture implicitly regulatory mechanisms that may not be measured by microarray experiments at the metabolite level.

Third, there is a realization that a model can only describe some properties of the real gene expression networks. SWNI is an unavoidably biased to single gene perturbation and steady-sate expression data for uncovering gene regulations. It emphasizes the most confident connections in the networks, leaving out other aspects that are relevant for the purpose of the study. In fact, the performance of all algorithms on the simulated data sets are biased as they are based on different theories. High predictive accuracy of SWNI can be some due to the linear ODE are also used to generate the simulated gene expression data in this case. Meanwhile, the performance of other algorithms can be further improved by modifying their parameters. For example, the NIR algorithm can be affected by varying the parameter *k* that each gene can be regulated at most by other *k* genes. However, as the best fitted parameters are chosen for all the algorithms separately and noise is added to the simulated data, our data sets should not affect the results considerably and the reported performance of them should not be far from the true one.

Further, the best performance versus run-time are achieved by SWNI approach, while the traversal problem of NIR causes both large time complexity and space complexity. Our numerical experiments suggest that when the number of genes is more than 200 or so and if the average degree is smaller than 10 (a reasonable estimate for real biological networks), our algorithm shows better efficiency. Besides, proper parallel computing algorithm will be designed to solve larger-scale problems because a prohibitive amount of computer time is required on a classical single-CPU computer (Zomaya 2006). Thus, the study on developing and utilizing high performance computing system enables completing the gene network detection task in a reasonable time.

Acknowledgements

The authors are grateful to Ruisheng and Wang for helping to improve the methods. This work was supported by Shanghai Leading Academic Discipline Project under Grant No. J50103.

References

Albert R. and Barabasi A. L. 1999 Emergence of scaling in random networks. *Science* **286**, 509–512.

Albert R. and Barabasi A. L. 2000 Topology of evolving networks: local events and universality. *Phys. Rev. Lett*. **85**, 5234–5237.

Amato R., Ciaramella A., Deniskina N., Del Mondo C., di Bernardo D., Donalek C. *et al*. 2006 A multi-step approach to time series analysis and gene expression clustering. *Bioinformatics* **22**, 589–596.

Basso K., Margolin A. A., Stolovitzky G., Klein U., Dalla-Favera and Califano A. 2005 Reverse engineering of regulatory networks in human B cells. *Nat. Genet*. **37**, 382–390.

Beal M. J., Falciani F., Ghahramani Z., Rangel C. and Wild D. L. 2005 A Bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics* **21**, 349–356.

Butte A. J. and Kohane I. S. 1999 Unsupervised knowledge discovery in medical databases using relevance networks. In *Fall symposium* (ed. N. Lorenzi), pp. 711-715. American Medical Informatics Association. Hanley and Belfu, Washington, USA.

Chen L. and Aihara K. 2002 Stability of genetic regulatory networks with time delay. *IEEE Trans. Circuits Syst*. **49**, 602–608.

Chen P. C. 2004 A computational model of a class of gene networks with positive and negative controls. *BioSystems* **73**, 13–24.

Davis J. and Goadrich M. 2006 The relationship between precision-recall and ROC curves. In proceedings of the 23rd international conference on machine learning, pp. 233–240. ACM, New York, USA.

De Jong H. 2002 Medeling and simulation of genetic regulatory systems: a literature review. *J. Comput. Biol*. **9**, 67–103.

De la Fuente A., Brazhnik P. and Mendes P. 2002 Linking the genes: inferring quantitative gene networks from microarray data. *Trends Genet*. **18**, 295–298.

De la Fuente A., Bing N., Hoeschele I. and Mendes P. 2004 Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics* **20**, 3565–3574.

Faith J. J., Hayete B., Thaden J. T., Mogno I., Wierzbowski J. *et al*. 2007 Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles. *PLoS Biol*. **5**, 54–66.

Featherstone D. E. and Broadie K. 2002 Wrestling with pleiotrophy: genomic and topological analysis of the yeast gene expression network. *Bioessays* **24**, 267–274.

Featherstone D. E., Rushton E. and Broadie K. 2005 Developmental regulation of glutamate receptor field size by nonvesicular glutamate release. *Nat. Neurosci*. **5**, 141–146.

Friedman N., Nachman I. and Pe'er D. 2000 Using Bayesian networks to analyze gene expression data. *J. Comput. Biol*. **3**, 601–620.

Gardner T., di Bernardo D., Lorenz D. and Collins J. 2003 Inferring genetic networks and identifying compound mode of action via expression profiling. *Science* **301**, 102–105.

Gardner T. S. and Faith J. 2005 Reverse-engineering transcription control networks. *Phys. Life Rev*. **2**, 65–88.

Iba H. and Mimura A. 2002 Inference of a gene regulatory network by means of interactive evolutionary computing. *Inform. Sci*. **145**, 225–236.

Kauffman S. 1974 The large scale structure and dynamics of gene control circuits: an ensemble approach. *J. Theor. Biol*. **44**, 167–190.

Liang S., Fuhrman S. and Somogyi R. R. 1998 A general reverse engineering algorithm for inference of genetic network architectures. *Pac. Symp. Biocomput*. **3**, 18–29.

Margolin A. A. and Califano A. 2007 Theory and limitations of genetic network inference from microarray data. *Ann. N. Y. Acad. Sci*. **1115**, 51–72.

Mendes P., Sha W. and Ye K. 2003 Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics* **19**, 22–29.

Mendoza L. Thieffry D. and Alvarez-Buylla E.R. 1999 Genetic control of flower morphogenesis in *Arabidopsis thaliana*: a logical analysis. *Bioinformatics* **15**, 593–606.

Schumacher M., Binder H. and Gerds T. 2007 Assessment of survival prediction models based on microarray data. *Bioinformatics* **23**, 1768–1774.

Shieh G. S., Chen C., Yu C. and Huang J. 2008 Inferring transcriptional compensation interactions in yeast via stepwise structure equation modeling. *BMC Bioinformatics* **9**, 1471–2105.

Soranzo N., Bianconi G. and Altafini C. 2007 Comparing association network algorithms for reverse engineering of large-scale gene regulatory networks: synthetic versus real data. *Bioinformatics* **23**, 1640–1647.

Styczynski M. P. and Stephanopoulos G. 2005 Overview of computational methods for the inference of gene regulatory networks. *Comp. Chem. Eng*. **29**, 519–534.

van Someren E. P. Wessels F. A, Backer E. and Reinders M. J. T. 2001 Robust genetic network modeling by adding noisy data. *Proc. IEEE-EURASIP Workshop on nonlinear signal and image processing*. Baltimore, Maryland, USA.

Wang Y., Joshi T. and Zhang X. S., Xu D. and Chen L. 2006 Inferring gene regulatory networks from multiple microarray datasets. *Bioinformatics* **22**, 2413–2420.

Watts D. J. and Strogatz S. H. 1998 Collective dynamics of 'small-worldâ networks. *Nature* **393**, 440–442.

Yeung M., Tegner J. and Collins J. 2002 Reverse engineering gene networks using singular value decomposition and robust regression. *Proc. Natl Acad. Sci. USA* **99**, 6163–6168.

Yu J., Smith V. A., Wang P. P., Hartemink A. J. and Jarvis E. D. 2004 Advances to bayesian network inference for generating causal networks from observational biological data. *Bioinformatics* **20**, 3594–3603.

Zak D. E., Gonye G. E., Schwaber J. S. and Doyle F. J. 2003 Importance of input perturbations and stochastic gene expression in the reverse engineering of genetic regulatory networks: insights from an identifiability analysis of an in silico network. *Genome Res*. **13**, 2396–2405.

Zomaya A. Y. 2006 *Parallel computing for bioinformatics and computational biology: models, enabling technologies and case studies,* 1st edition. Wiley, New Jersey, USA.

Received 8 May 2009, in revised form 19 November 2009; accepted 30 November 2009

Published on the Web: 19 April 2010